	<pre>cv.waitKey(0) img = cv.cvtColor(f, cv.COLOR_BGR2RGB) fig, ax = plt.subplots() ax.imshow(img) ax.set_title('Original Image') plt.show() gamma_list = [0.2, 0.8, 1.2, 2]</pre>	
	<pre>for gamma in gamma_list: t = np.array([(p/255)**gamma*255 for p in range(0,256)]).astype(np.uint8) g = cv.LUT(f, t) title = 'y = ' + str(gamma) fig, ax = plt.subplots() ax.set_title(title) ax.plot(t) cv.imshow('Image', g) cv.waitKey(0)</pre>	
	<pre>fig, ax = plt.subplots() img = cv.cvtColor(g, cv.COLOR_BGR2RGB) ax.imshow(img) title1 = 'Image with gamma correction (y = ' + str(gamma) + ')' ax.set_title(title1) plt.show()</pre> cv.destroyAllWindows()	
	Original Image 100 - 100 - 150 - 250 - 350 - 350 -	
	$ \begin{array}{ccccccccccccccccccccccccccccccccccc$	
	100 - 50 - 100 - 150 - 200 - 250 - 100 - 150 - 200 - 250 - 100 - 150 - 200 - 250 - 100 - 150 - 200 - 250 - 100 -	
	50 - 100 - 150 - 200 - 200 300 400 500 600 700	
	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
	50 - 100 150 200 250 Image with gamma correction (γ = 0.8) 100 - 150	
	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	
	200 - 150 - 100 -	
	Image with gamma correction (γ = 1.2) 100	
	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	
	150 - 100 - 50 - 100 - 150 - 200 - 250 -	
	50 - 100 - 150 - 200 - 250 - 350 - 400 -	
	import numpy as np import cv2 as cv import matplotlib.pyplot as plt f = cv.imread(r'spider.png', cv.IMREAD_GRAYSCALE) assert f is not None t1 = np.linspace(0,100,51) t2 = np.linspace(101, 255, 150)	
	<pre>t3 = np.linspace(255,255,55) t = np.concatenate((t1, t2, t3), axis=0).astype(np.uint8) fig, ax = plt.subplots() ax.plot(t) ax.set_title('Intensity Transformation') ax.set_xlabel('Input Intensity') ax.set_ylabel('Ouput Intensity') assert len(t) == 256 g = cv.LUT(f, t)</pre>	
	<pre>cv.namedWindow('Image', cv.WINDOW_AUTOSIZE) cv.imshow('Image', f) cv.waitKey(0) cv.imshow('Image', g) cv.waitKey(0) cv.destroyAllWindows() fig, ax = plt.subplots(2,1,figsize = (12,10)) plt.subplot(1,2,1) plt.imshow(cv.cvtColor(f, cv.CoLOR_BGR2RGB)) plt.title('Original Image')</pre>	
	plt.subplot(1,2,2) plt.imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB)) plt.title('Intesity Transformation') plt.show() Intensity Transformation 250 200	
	150 - 100 - 150 200 250 Input Intensity	
	Original Image Intesity Transformation 100 - 200 - 300 - 40	
	import numpy as np import cv2 as cv import matplotlib.pyplot as plt f = cv.imread('shells.tif',cv.IMREAD_GRAYSCALE) assert f is not None hist_f = cv.calcHist([f], [0], None, [256], [0,256]) g = cv.equalizeHist(f)	
	<pre>hist_g = cv.calcHist([g], [0], None, [256], [0,256]) fig, ax = plt.subplots(2,1,figsize = (12,4)) plt.subplot(1,2,1) plt.plot(hist_f) plt.title('Histogram of Original Image') plt.subplot(1,2,2) plt.plot(hist_g) plt.title('Histogram of Equilized Image') plt.show()</pre> fig, ax = plt.subplots(2,1,figsize = (12,8))	
	plt.subplot(1,2,1) plt.imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB)) plt.title('Original Image') plt.subplot(1,2,2) plt.imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB)) plt.title('Equilized Image') plt.show() Histogram of Original Image Histogram of Equilized Image 17500 15000 Histogram of Equilized Image	
	12500 - 10000 - 7500 - 5000 - 2500 - 0 50 100 150 200 250	
	Original Image 100 - 20	
	300 - 300 - 400 - 400 - 400 - 400 - 100 200 300 400	
	<pre>import numpy as np import cv2 as cv import matplotlib.pyplot as plt f = cv.imread('zion_pass.jpg',cv.IMREAD_COLOR) assert f is not None hsv = cv.cvtColor(f, cv.CoLOR_BGR2HSV) h, s, v = cv.split(hsv) t1 = np.linspace(0,100,51)</pre>	
	<pre>t2 = np.linspace(101, 255, 150) t3 = np.linspace(255,255,55) t = np.concatenate((t1, t2, t3), axis=0).astype(np.uint8) assert len(t) == 256 s_ = cv.LUT(s, t) fig, ax = plt.subplots() ax.set_title('Saturation Transformation') ax.plot(t)</pre>	
	<pre>final_hsv = cv.merge((h, s_, v)) g = cv.cvtColor(final_hsv, cv.COLOR_HSV2BGR) cv.namedWindow('Image', cv.WINDOW_AUTOSIZE) cv.imshow('Image', f) cv.waitKey(0) cv.imshow('Image', g) cv.waitKey(0) cv.waitKey(0) cv.waitKey(0) cv.destroyAllWindows()</pre> fig, ax = plt.subplots(2,1,figsize = (12,10)) plt.subplot(1,2,1)	
]:	plt.subplot(1,2,1) plt.imshow(cv.cvtColor(f, cv.CoLOR_BGR2RGB)) plt.title('Original Image') plt.subplot(1,2,2) plt.imshow(cv.cvtColor(g, cv.CoLOR_BGR2RGB)) plt.title('Saturation Enhanced Image') Text(0.5, 1.0, 'Saturation Enhanced Image') saturation transform γ = 0.2	
	200 - 150 - 100 - 50 -	
	0 50 100 150 200 250 Original Image 100 - 200 - 300 - 300 - 300 - 400	
]:	import numpy as np import cv2 as cv import matplotlib.pyplot as plt f = cv.imread('zion_pass.jpg',cv.IMREAD_COLOR) assert f is not None	
	<pre>hsv = cv.cvtColor(f, cv.CoLoR_BGR2HSV) h, s, v = cv.split(hsv) t = np.array([(p/255)**gamma*255 for p in range(0,256)]).astype(np.uint8) h_ = cv.LUT(h, t) fig, ax = plt.subplots() ax.set_title('Hue Transformation') ax.plot(t) final_hsv = cv.merge((h_, s, v))</pre>	
	<pre>g = cv.cvtColor(final_hsv, cv.CoLoR_HSV2BGR) cv.namedWindow('Image', cv.WINDOW_AUTOSIZE) cv.imshow('Image', f) cv.waitKey(0) cv.imshow('Image', g) cv.waitKey(0) cv.waitKey(0) cv.destroyAllWindows() fig,ax = plt.subplots(2,1,figsize = (12,10)) plt.subplot(1,2,1)</pre>	
	plt.imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB)) plt.title('Original Image') plt.subplot(1,2,2) plt.imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB)) plt.title('Hue changed Image') Text(0.5, 1.0, 'Hue changed Image') Hue Transformation	
	150 -	
	100 - 50 - 0 50 100 150 200 250	
	50 -	

import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

f = cv.imread(r'spider.png', cv.IMREAD_GRAYSCALE)
assert f is not None